[ROK Input Paper to the IMO e-Navigation Correspondence Group]

**Introduction**

The Republic of Korea believes that the successful implementation of e-navigation needs to consider software quality. Associated with the e-navigation implementation strategy, various systems and services will be the result of new software development(s). One example would be a where real-time tidal or water level information is:

- measured by a water level sensor

- broadcast from an AIS Base Station as an AIS Application-Specific Message

- received by an AIS transceiver onboard a vessel or ashore

- incorporated into a “Dynamic ENC” that is displayed on a shipborne or shore-based electronic chart display (e.g., ECDIS, INS, or a Portable Piloting Unit).

In order for this to occur, high-quality software that is both stable and complete will need to be developed and installed. To do this for both shipborne and shore-based systems – in a harmonized manner -- practical guidelines will need to be developed and agreed.

In many respects, the need for quality software development has been identified by the e-navigation gap analysis process. High-quality software will also contribute to the effectiveness of the Common Maritime Data Structure. Software quality assurance guidelines should be included in the e-navigation implementation process under consideration by IMO.

**The Concept of Software Quality**

As it relates to software engineering, software quality has two aspects**:**

1. Softwarefunctional quality reflects how well it complies/conforms to specific user requirements, especially functional requirements (e.g., e-navigation).

2. Softwarestructural quality refers to how it meets non-functional requirements, such as robustness, maintainability, security or correctness of data delivery.

Structural quality is evaluated through the analysis of the software inner structure and, source code. In effect, it relates to how its architecture adheres to sound principles of software architecture. In contrast, functional quality is typically evaluated through software testing.

Measuring software quality involves two management processes:

Risk Management **-** Risk is the uncertainty which something may occur, and a corresponding potential for loss in the future. Software error/failure might cause human fatalities more than inconvenience. The causes range from poorly designed user interfaces to direct programming errors.

For example, In the United States, the Aircraft Certification Service of the Federal Aviation Administration (FAA) provides software programs, policy, guidance and training, focus on software and Complex Electronic Hardware that has an effect on the airborne “products “(e.g., an aircraft) and Air Traffic Control Services. This will likely apply to shipborne and shore-based e-navigation systems.

Cost Management **-** In any field of engineering, an application with good structural software quality costs less to develop, maintain, and modify according to user needs. If the application structural quality is poor, it can lead to significant problems in terms of corrupted data, application outages, security breaches, and performance problems.

ISO/IEC 9126-1 classifies software quality in a structured set of characteristics and sub-characteristics:

**Functionality** - *A set of attributes that bear on the existence of a set of functions and their properties. The functions are those that satisfy stated or implied needs.*

* + [Suitability](http://en.wikipedia.org/w/index.php?title=Suitability&action=edit&redlink=1)
  + Accuracy
  + [Interoperability](http://en.wikipedia.org/wiki/Interoperability)
  + [Security](http://en.wikipedia.org/wiki/Computer_security)
  + Functionality Compliance

[**Reliability**](http://en.wiktionary.org/wiki/reliability) - *A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time.*

* + Maturity
  + [Fault Tolerance](http://en.wikipedia.org/wiki/Fault_Tolerance)
  + Recoverability
  + Reliability Compliance

[**Usability**](http://en.wikipedia.org/wiki/Usability) - *A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users.*

* + Understandability
  + [Learnability](http://en.wikipedia.org/wiki/Learnability)
  + [Operability](http://en.wikipedia.org/wiki/Operability)
  + Attractiveness
  + Usability Compliance

[**Efficiency**](http://en.wikipedia.org/wiki/Algorithmic_efficiency) - *A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions.*

* + Time Behavior
  + Resource Utilization
  + Efficiency Compliance

[**Maintainability**](http://en.wikipedia.org/wiki/Maintainability) - *A set of attributes that bear on the effort needed to make specified modifications.*

* + Analyzability
  + Changeability
  + Stability
  + [Testability](http://en.wikipedia.org/wiki/Testability)
  + Maintainability Compliance

[**Portability**](http://en.wikipedia.org/wiki/Software_portability) - *A set of attributes that bear on the ability of software to be transferred from one environment to another.*

* + Adaptability
  + Installability
  + Co-Existence
  + Replaceability
  + Portability Compliance

Each quality sub-characteristic (e.g., adaptability) is further divided into detail attributes to be measured.

Other ISO/IEC standards related to software quality that should be considered in association with the e-navigation implementation process include:

ISO 14915: Software ergonomics for multimedia user interfaces

ISO 18789: Ergonomic requirements and measurement techniques for electronic visual displays

ISO/IEC 12207: Systems and software engineering -- Software life cycle processes

ISO/IEC 15026: Information technology -- System and software integrity levels

ISO/IEC 14598: Information technology -- Software product evaluation

ISO 9241: Ergonomics of human-system interaction

ISO/IEC 18019: Software and system engineering - Guidelines for the design and preparation of user documentation for application software

ISO/IEC 26513: Systems and software engineering - Requirements for testers and reviewers of user documentation

ISO/IEC 26514: Systems and software engineering -- Requirements for designers and developers of user documentation

ISO/IEC/IEEE 26512: Systems and software engineering -- Requirements for acquirers and suppliers of user documentation

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_